Sun™ ONE Integration Server, EAI 3.1 Release Notes

Version 3.1

Part Number 817-4042-10

September 2003

These release notes contain important information available at the time of the release of Sun™ Open Net Environment (Sun ONE) Integration Server, Enterprise Application Integration Edition, Version 3.1 (Integration Server, EAI). New features and enhancements, known limitations and problems, technical notes, and other information are addressed here. Read this document before you begin using this release of Integration Server, EAI.

The most up-to-date version of these release notes can be found at the Sun ONE documentation web site: http://docs.sun.com/db/coll/S1_IntegrationServer_EAI_31

Check the web site prior to installing and setting up your software and then periodically thereafter to view the most up-to-date release notes and manuals.

These release notes contain the following sections:

- Revision History
- About Integration Server, EAI, Version 3.1
- What's New in Integration Server, EAI Version 3.1
- Hardware and Software Requirements
- Installation Notes
- Errata and Integration Server, EAI Documentation Updates
- Compatibility Issues
- Known Issues
- How to Report Problems and Provide Feedback
- Additional Sun Resources

Revision History

Table 1 Revision History

Date	Description of Changes
September 16, 2003	Initial release of these release notes

About Integration Server, EAI, Version 3.1

Sun ONE Integration Server, EAI 3.1 is a software product designed for enterprises that need to integrate packaged, custom, legacy, and new applications. Integration Server, EAI provides integration for Web services through SOAP messaging, which can export a service definition via WSDL to UDDI registries or other SOAP clients.

Sun ONE Integration Server, EAI 3.1 makes it possible for companies to integrate core business processes with multiple applications, running on multiple operating systems, across multiple communication protocols within the enterprise. Companies benefit from the automation of business processes across distributed heterogeneous information systems with increased productivity and efficiency.

This release of Sun ONE Integration Server, EAI 3.1 features new support for ebXML messaging, which is a cutting-edge technology for web-services and e-business collaboration that provides secure and reliable messaging between trading partners.

Bundled with this release are the following products

- Sun ONE Unified Development Server 5.1.4.

  This is a patch release based upon Sun ONE Unified Development Server 5.1.3. This software must be installed before you install Sun ONE Integration Server, EAI 3.1.
- Sun ONE Message Queue 3.0.1 SP 2 Professional Edition
- Sun ONE Connector Builder 2.0
- Sun ONE Integration Server Secure Trading Agent 1.0

As part of Sun ONE Integration Server, EAI 3.1, Sun Microsystems is providing customers with five additional licenses of the Sun ONE Integration Server, Secure Trading Agent (STA) 1.0. This lightweight client software uses key Web Services standards to enable quick and easy connections with trading partners.

Customers can choose to:

a. Use all five of the licenses for internal STA deployments at designated IP addresses

b. Distribute some or all of the licenses to trading partners for their use

Your media kit includes a card containing the URL where you can download Sun ONE Integration Server, Secure Trading Agent (STA) 1.0.

What's New in Integration Server, EAI Version 3.1

New features in Integration Server, EAI 3.1 include:

- Proxy Performance Enhancements
- Support for ebXML Messaging
- ebXML Example Application
- IIOP Configuration Enhancements
- Changes to the Configure New Engine and Reconfigure Engine Dialog Boxes
- Workshop Localization Support
- New FNscript Command SetHttpProxy
- Enhancements to the Import WSDL Dialog

Proxy Performance Enhancements

In internal long-term performance benchmark tests, Sun ONE Integration Server, EAI 3.1 proxies were measured to be over twice as fast on average than Integration Server, EAI, 3.0.1 proxies.

Support for ebXML Messaging

Integration Server, EAI proxies can be configured to communicate with trading partners through ebXML.

A proxy can establish a session with a process engine, and associate an activity of a process to an action of a partner agreement using Secure Trading Agent to send ebXML. A proxy can also poll Secure Trading Agent for received ebXML messages to start processes, or complete process activities registered with the process engine by sending a message.

Three basic proxy service configurations are possible, two for inbound message handling, and one for outbound message handling:

- ebXML *Initiator* service

  This inbound configuration is used to create a new Integration Server, EAI process by receiving a message from your trading partner through Secure Trading Agent.

- ebXML *Sender* service

  This outbound configuration is used to complete an Integration Server, EAI process activity by sending a message through Secure Trading Agent to your trading partner.

- ebXML *Receiver* service

  This inbound configuration is used to complete an Integration Server, EAI process activity by receiving a message from your trading partner through Secure Trading Agent.

Each proxy must be configured appropriately as an Initiator, Sender, or Receiver service, depending on how it communicates with Secure Trading Agent and the process engine.

Defining a proxy service for ebXML is similar to defining a SOAP proxy service using the FNscript command `AddServiceDefinition`. For more information about ebXML proxy services, see the *iPlanet Integration Server Backbone System Guide*.

A SunSolve article has been created to provide last minute information about ebXML functionality in Integration Server, EAI. For more information, see article number 71455 at this location: http://sunsolve.sun.com/pub-cgi/retrieve.pl?doc=fsunone%2F71455.

ebXML Example Application

The *FORTE_ROOT*`/install/examples/fusion/ebxml` directory contains a sample application that simulates an ebXML message exchange between two trading partners and a process registered with the Integration Server, EAI engine.

This example ensures that your environment is set up to receive messages from a remote installation of Secure Trading Agent with your local installation of Secure Trading Agent. An Integration Server, EAI proxy service, called an Initiator, polls Secure Trading Agent for received messages. When an ebXML message matching the proxy service definition arrives, the proxy delivers information to the Integration Server, EAI engine to start a process. If the message is valid, a second ebXML proxy service, called a Sender, sends a corresponding message back to the remote installation of Secure Trading Agent from an activity in the process.

For instructions on how to install and run the example application, see the `readme.htm` file in the example directory.

IIOP Configuration Enhancements

When you create a new Integration Server, EAI engine using the Integration Server Console (formerly iIS console), you can create an IIOP server partition that starts up automatically upon engine startup.

The default locale for this IIOP server is now `en_us.utf8`. This codeset improves support for Java IIOP clients.

Using the Cscript `startengine` command, you can now launch the IIOP server partition when the process engine is launched, if you have configured it to be part of the process engine. The syntax for `startengine` remains the same.

Note: You can still use the `IIOPServer <Start> <Stop> <Show>` command to start and stop the IIOP Server.

Changes to the Configure New Engine and Reconfigure Engine Dialog Boxes

When creating a new Integration Server, EAI engine with Integration Server console, the following engine startup options are available:

- ability to add an IIOP server as an optional part of the process engine

  This enables automatic startup of the IIOP server when the process engine is launched.
- localization override options for all process engine components
- log flag override options for all process engine components
- memory flag and stack size override options all process engine components

- default memory values for all Integration Server engine components that take into account production-level process load conditions

The Configure New Engine and Reconfigure Engine dialog boxes now have separate tabs for engine, governor, database service, and iiop server components for the engine. The fields on each of these tabs allow you to override engine startup parameters on a component-by-component basis. The engine startup parameters you can override are described in Table 2.

Table 2 New Startup Parameters for Engine Components

Parameter	Description
Memory flags	Enables you to override engine-wide memory flags for a specific engine component. The default setting for the memory flags parameter is different for each component. For instance, the memory flag setting for an IIOP server is (n:8192,i:8192,x:32768). This means the minimum value is 8192 bytes, the incremental value is 8192 bytes, and the maximum value is 32768 bytes. If your production system handles a large workload, you will most likely want to increase the value of Memory flags above the minimum, to achieve reasonable throughput. For information on UDS memory flags, refer to the online help or the *iPlanet UDS System Management Guide*.
Stack size	Enables you to override the engine-wide stack size for a particular partition. Default stack sizes for engine components are as follows: • Engine Unit - 128000 bytes • Database Service - 128000 bytes • Governor - 64000 bytes • IIOP Server - 96000 bytes
Locale	Enables you to override localization settings for a specific partition. For example, if an engine partition has to communicate with external clients that use a different codeset or locale than the process engine, you could configure that engine unit to use the external clients' localization settings.
Log flags	Specifies log flags for internal debugging traces. For more information about setting log flags and log flag syntax, see the chapter "Troubleshooting," in the *iPlanet Unified Development Server System Management Guide*.

In the Engine tab, the default value for the memory-resident process limit parameter has been increased to 50000. The Resident Process Limit determines the number of process instances retained in memory. When the number of memory-resident process instances exceeds this limit, the engine swaps out dormant process instances, allowing for execution of newly activated process instances.

A number of new keywords have been added to the engine configuration file, to support the new partition startup parameter values:

- `DBSERVICE_LOCALE`
- `DBSERVICE_LOG_FLAGS`
- `GOVERNOR_LOCALE`
- `GOVERNOR_LOG_FLAGS`
- `GOVERNOR_MEMORY_FLAGS`
- `GOVERNOR_STACK_SIZE`
- `IIOP_LOCALE`
- `IIOP_LOG_FLAGS`
- `IIOP_MEMORY_FLAGS`
- `IIOP_STACK_SIZE`
- `UNIT_LOCALE`
- `UNIT_LOG_FLAGS`

For more information about configuring an Integration Server, EAI engine, see "Configuring an Engine" in the Integration Server Process System Management online help.

Workshop Localization Support

In previous releases, processes, activities, and timers were mapped under their class names. This made localization of these object names difficult. A label name field has been added to many object definition windows in the process development workshops. The name specified in this field will be displayed and preserved in the native locale set for the process development workshop.

New FNscript Command SetHttpProxy

This release includes a new FNscript command `SetHttpProxy` to enable the use of web proxies. Web proxies are objects that act as both clients and servers in order to make requests and receive responses on behalf of other clients.

The most common use for web proxies is to cache responses for repeated requests. If a client makes the same request more than once the response can be fetched from a local cache (assuming the same response is appropriate for subsequent requests), instead of the client having to reconnect to an external server. This can improve response time and make more efficient use of network resources.

In certain environments, it may be necessary to use proxy servers for communication. The `SetHttpProxy` command specifies the HTTP Proxy web server settings for a proxy. If set at the backbone level, `SetHttpProxy` sets the default HTTP Proxy settings for all Integration Server, EAI proxies in the backbone. For more information on how to use this command, see the *Integration Server Backbone System Guide*.

Enhancements to the Import WSDL Dialog

The Import WSDL dialog, available from the Service Information dialog in the Application Dictionary Workshop, now imports output parameters in WSDL files and suggests possible application dictionary item names for both input and output parameters. You must map these parameters to the application dictionary items. The dialog also supports the import of input and output parameters in SOAP headers.

Bugs Fixed in Unified Development Server, UDS 5.1.4

Below is a short description of the most important bugs fixed in Unified Development Server, 5.1.4, the companion software for Integration Server, EAI 3.1. This release is a patch release built upon Unified Development Server 5.1.3.

Table 3 Fixed Bugs in Unified Development Server 5.1.4

Bug Number	Description
4854459	The WSDL file is generating all structs and arrays within the Service Object's (SO) namespace instead of a namespace that includes just the project name.
4899032	Install which allowed spaces in $FORTE_ROOT caused problems with fcompile.
4905147	If a ServiceName is not found, fortensapi will return a message with the original input which could contain a "malicious tag" and script that would then be run on the webserver.
4908948	Missing Oracle DB Resource Manager on HP/UX

Bugs Fixed in Integration Server, EAI 3.1

Below is a short description of the most important bugs fixed in Integration Server, EAI 3.1.

Table 4 Fixed Bugs in Integration Server, EAI 3.1

Bug Number	Description
56408	Process definition registrations deleted in error (AIX).
4630616	Handling of UTF-8 characters for XMLData class is incorrect.
4634018	Large Unicode process attributes can cause engine shutdown.
4642909	Proxy limit on types of JMS messages is not documented. See Errata and Integration Server, EAI Documentation Updates section.
4643160	Illegal SQLStatement on DB2 operand “=NULL” not compatible, should be “IS NULL”.
4644359	Example command to start Dump/Restore utility is incorrect in the IS EAI documentation. See Errata and Integration Server, EAI Documentation Updates section.
4695164	Saving large xmldata process attributes is very inefficient.
4698421	Wrong keyword shown in JMS configuration example.
4704871	SetAplRetry values expressed in milliseconds, not seconds as documentation says.
4704871	SetAplRetry values expressed in milliseconds, not seconds as documentation says.
4710927	Scope errors generated when attempting to access XMLDOM2 library from process definition.
4712188	Loading supplier plan library can generate errors and disable the process engine.
4713915	After engine crash, JMS enabled proxy removes unprocesses messages from JMS broker.
4729939	Conductor client crashes after session retry count reaches 0.
4730650	Conductor client can crash while calling getState().
4756677	Proxy config setting required to suppress ActivityExists state document.
4783686	Process and activity left inaccessible after engine fail-over.
4785575	Empty XML attribute values get corrupted.
4810499	Proxy under load can experience JMS timeouts.
4816495	Proxy conversion error processing valid XML file.

Hardware and Software Requirements

For information about hardware and software requirements for this release of Integration Server, EAI, refer to the platform certification matrix at the following location:
http://docs.sun.com/db/coll/S1_IntegrationServer_EAI_31

Support for SOAP v1.1

Integration Server, EAI 3.1 conforms with SOAP v1.1, as defined by the W3C (http://www.w3.org/TR/2000/NOTE-SOAP-20000508), with the following limitations:

- Integration Server, EAI expects SOAP messages in the SOAP-RPC style, not the document style
- Integration Server, EAI does not perform type checking for input and output parameters
- Datatype conversion is limited to corresponding process engine attribute datatypes
- Complex parameters (arrays and structures) can only be mapped to XmlData process attributes
- Headers are ignored unless the MustUnderstand attribute is set, in which case an error is generated
- SOAP header and SOAP message chaining are not supported; limited header I/O is supported

Support for SOAP with this version of Integration Server, EAI has been certified for interoperability with the SOAP support available with Sun ONE Unified Development Server 5.1.3. For more information on SOAP support for this release, refer to Chapter 3 of the *Integration Server Backbone System Guide*.

Installation Notes

Before you install Sun ONE Integration Server, EAI 3.1, you must install Sun ONE Unified Development Server 5.1.4. This release is a patch release of Unified Development Server 5.1.3 that includes bug fixes necessary to operate Integration Server, EAI 3.1. This patch release is included in the media you received when you purchased Integration Server, EAI 3.1.

Unpacking the Distribution Files

This section describes how to prepare the downloaded files for installation.

Table 5 lists the distribution files available with this release.

Table 5 Integration Server, EAI Distribution Files

Distribution File	Platform
`axp_osf_eai.tar`	Compaq Tru64 UNIX
`hp_ux_eai.tar`	HP 9000
`ibm_aix_eai.tar`	IBM RS/6000
`pc_w32_eai.exe`	Windows platforms
`sun_sol_eai.tar`	Sun SPARC

After downloading the distribution, you need to unpack the files before beginning your installation. The instructions for unpacking the files vary according to platform. Refer to the following sections for instructions on unpacking the distribution files.

Windows Platforms

For Windows platforms, the distribution file, `pc_w32_eai.exe` is a compressed file. To extract the files necessary for installation on Windows:

1. Create a target directory for the extraction inside a top-level local directory.

   For example, create an `eai\eai_3_1` directory.

2. Extract the contents of this `pc_w32_eai.exe` file to the `eai_3_1` directory

You are now ready to proceed with your installation on Windows, as described in the *iPlanet Integration Server Installation Guide*.

When you extract Integration Server, EAI to the target directory on Windows, Sun ONE Message Queue 3.0.1 SP2 is extracted to a `messagequeue3_0_1` directory within that target directory. Documentation for Sun ONE Message Queue 3.0.1 SP2 can be found at the following location:

http://docs.sun.com/?p=/coll/S1_MessageQueue_301

UNIX Platforms

The Integration Server, EAI install script for UNIX platforms is based on an installation CD image that includes directories for each platform certified for this release. Download the Solaris distribution to your Sun SPARC server and then extract the distribution files:

```
% tar xvf sun_sol_eai.tar
```

This creates a `CDROM` directory. The `CDROM` directory contains an `ALL_UNIX` directory and a `SUN_SOL` directory.

You are now ready to proceed with your central server node installation on Solaris, as described in the *iPlanet Integration Server Installation Guide*. After you complete your central server node installation, you can then proceed with either a server node or runtime client node installation on the Windows and HP platforms.

When you extract Integration Server, EAI on Solaris, Sun ONE Message Queue 3.0.1 SP2 is extracted to a `messagequeue3_0_1.zip` archive within the `SUN_SOL` directory. Documentation for Sun ONE Message Queue 3.0.1 SP2 can be found at the following location:

http://docs.sun.com/?p=/coll/S1_MessageQueue_301

Errata and Integration Server, EAI Documentation Updates

Use the Integration Server, EAI 3.0 Documentation with Integration Server, EAI 3.1. The Integration Server, EAI 3.0 documentation contains iPlanet branding. You can find the documentation at the following location:

http://docs.sun.com/db/coll/S1_IntegrationServer_EAI_31

Documentation Errata

- In the readme for the ebXML example application located in the *FORTE_ROOT*`/install/examples/fusion/ebxml` directory, there is a section that is no longer relevant to the setup of the example application.

  In the subsection “Modifying the smplecpa.xml CPA” of the section “Setup Related to Secure Trading Agent”, you can ignore the “From the remote directory:” directions.

- In the section on configuring JMS proxies in the *Integration Server Backbone System Guide* it should state that the only JMS message body types (or classes) Integration Server, EAI proxies can handle are TextMessage and ByteMessage.
- On p. 127 of the *Integration Server Process System Guide*, the following example command is given to start the Dump/Restore utility from a UNIX, OpenVMS, or Windows NT command line:

  ```
  ftexec -fnict -fcons -fi bt:\$FORTE_ROOT\userapp\wfdrdump\c20\wfdrdu0
  ```

  This command is incorrect. The correct syntax for the dump command looks something like this:

  ```
  ftexec -fcons -fi bt:$FORTE_ROOT/userapp/wfdrwind/cl1/wfdrwi0 -d
  <directory>
  ```

  The -d option is now required, not optional

  Syntax for drdump command:

  ```
  drdump -e engine_name -d directory_name [-v data_file_name] [-t
  registration|state|history]
  ```

- The FNscript commands `AddToJvmArgs`, `RemoveFromJvmArgs`, `AddToJmsClasspath`, and `RemoveFromJmsClasspath` are documented in the *Integration Server Backbone System Guide* as backbone level commands, when they only work at the proxy/component level.

  Call the FNscript commands `AddToJvmArgs`, `RemoveFromJvmArgs`, `AddToJmsClasspath`, and `RemoveFromJmsClasspath` at the proxy/component level.

- The following supported FNscript commands are missing from the *Integration Server Backbone System Guide:*
  - `SetLocaleName`
  - `ClearLocaleName`
  - `SetLoggerFlags`
  - `ClearLoggerFlags`

  SetLocaleName and ClearLocaleName Commands

  Both Integration Server, EAI and the base Unified Development Server product have features for localization of your systems. These new FNscript commands allow you to configure an Integration Server, EAI backbone and its proxies for a specific locale from the set of locales supplied with these products:

  `SetLocaleName` *`locale`*

  The `SetLocaleName` command works at the backbone and proxy levels.

  NOTE: A locale set at the proxy level overrides one set at the backbone level.

The locale argument uses the same format as the `FORTE_LOCALE` environment variable. For a description of `FORTE_LOCALE`, see the manual *iPlanet UDS Building International Applications*. For a complete list of locales supplied with Sun ONE Integration Server, EAI 3.1 and Unified Development Server 5.1, see SunSolve support article 7737.

`ClearLocaleName`

This command clears the locale set at the backbone or proxy level.

NOTE: If a proxy's locale is cleared, the locale set at the backbone level will take effect the next time that proxy is started.

SetLoggerFlags and ClearLoggerFlags Commands

Logger flags generate low-level data for problem analysis or override default behavior. The following FNscript commands allow you to configure logger flags for an Integration Server, EAI backbone and its proxies:

`SetLoggerFlags` *`file_name(file_filter)[file_name(file_filter)...]`*

`SetLoggerFlags` sets logging at the backbone or proxy level.

NOTE: Logger flags set at the proxy level override those set at the backbone level.

The parameter for `SetLoggerFlags` requires the same format as the argument for the `-fl` command-line logger flag. For more information about logger flags, see the *UDS System Management Guide*. For trace flags and configuration flags specific to the Integration Server, EAI backbone and proxies, see SunSolve support article 6914.

`ClearLoggerFlags`

This command clears the logger flags at the backbone or proxy level.

NOTE: If a proxy's logger flags are cleared, any logger flags set at the backbone level will take effect the next time that proxy is started.

- In the Integration Server Process System Management online help, the description of the Log flags field for all engine component startup parameters in the Configure New Engine and Reconfigure New Engine dialog boxes is incorrect.

  Currently it states: Log flags

  Specifies which tables (current state and history log) are enabled and which object state changes are recorded if the history log is enabled. String contains any of the following entries, separated by commas:

  `[STATE] [,LOG] [,LOG_ACTIVITIES] [,LOG_ATTRIBUTES] [,LOG_LOCKS] [,LOG_PROCESSES] [,LOG_SESSIONS] [,LOG_TIMERS]`

  The history log can grow quite large, so set logging selectively. You might have to monitor, back up, and flush the history log tables on a regular basis.

It should state: Log flags

Specifies log flags for internal debugging traces.

For more information about setting log flags and log flag syntax, see the chapter "Troubleshooting," in the *iPlanet Unified Development Server System Management Guide*.

Documentation Updates

The *Integration Server Backbone System Guide* has been updated to include support for ebXML messaging. For more information about this Integration Server, EAI feature, see "What's New in Integration Server, EAI Version 3.1" on page 3.

The chapter "Proxy Concepts and Features" has been updated to include a conceptual overview of ebXML communication.

The chapter "Backbone System Configuration and Administration" has been updated to include backbone and proxy example configurations for ebXML communication.

The following FNscript commands have either been added or updated to include new functionality in Appendix A, "Fusion Script Commands", in the *Integration Server Backbone System Guide.*

```
AddListener
AddServiceDefinition
AddToJvmArgs
RemoveFromJvmArgs
RemoveServiceDefinition
SetCSession
SetHttpProxy
SetPort
SetProtocol
ShowServiceDefinition
```

The Integration Server online help has been updated to include new engine configuration features. For more information about this Integration Server, EAI feature, see "What's New in Integration Server, EAI Version 3.1" on page 3.

Documentation References to the Platform Matrix

The *iPlanet UDS System Installation Guide* and other manuals provide an outdated URL to the Integration Server, EAI platform matrix. The correct URL to the platform matrix is:

http://docs.sun.com/db/coll/S1_IntegrationServer_EAI_31

New Article Database

This release of Integration Server, EAI introduces a new article database for support articles. In the previous release, technotes were on the iPlanet Knowledge base. All technotes formerly posted to the iPlanet Knowledge base can now be found at the SunSolve support site.

To find an article when you know the article number, go to the following location:

http://sunsolve.sun.com

Once there, enter the article number in the Search SunSolve field and click the Search button. Your article should appear in the results page.

Documentation in Text Format

You can download a full zipped set of Sun ONE Integration Server, EAI 3.0 documentation in ASCII file format.

We provide these files for individuals who are unable to use the PDF files as delivered. The ASCII files can be used with assistive technologies that enable specialized input and output capabilities and make the documentation accessible by people with disabilities.

Compatibility Issues

Websphere MQ 5.2 with IBM AIX 5.1

Customers who want to use Websphere 5.2 on IBM AIX 5.1 for use with Integration Server JMS proxies must use JDK/JRE version 1.3 to start proxies.

Microsoft Visual Basic .Net

In the newest version of Microsoft's Visual Basic product (Visual Basic .Net), the implementation of the declaration of session objects has been changed.

Code examples in the ActiveX chapter of the *Process Client Programming Guide* will not work with Visual Basic .Net. Likewise, the ActiveX example application, Expense Reporting, shipped with Integration Server, EAI 3.1 will not work with Visual Basic .Net. Numerous changes to the implementation of the Visual Basic environment require that you change ActiveX object declarations considerably.

The following short code examples demonstrate the previous way to declare an ActiveX session object and the new way to declare an ActiveX session object

Visual Basic 6 and earlier declaration of an ActiveX object

```
Dim DirectorySO As Object
Set DirectorySO = CreateObject("CONDUCTORCLIENT.WFDirectoryServiceCtl.1")
```

Visual Basic .Net declaration of an ActiveX object

```
Dim DirectorySO As ConductorClientLib.WFDirectoryService
DirectorySO = New ConductorClientLib.WFDirectoryService()
```

For customers with previous versions of ActiveX clients, Sun recommends using a previous version of Visual Basic to compile session code.

If you want to upgrade your ActiveX applications to use Visual Basic .Net, refer to Microsoft's Visual Basic documentation for information on how to migrate old ActiveX applications.

The ActiveX libraries included with Integration Server, EAI 3.1 are supported with the use of previous versions of Visual Basic and the new version of Visual Basic .Net.

Known Issues

This section contains a list of the more important known issues at the time of the Integration Server, EAI 3.1 release.

Table 6 Known Issues with Integration Server, EAI 3.1

Bug Number	Description
4900080	ActiveX Expense Reporting example (plus doc) needs updating if customers decide to upgrade to Visual Basic .Net from previous versions. **Workaround:** See the Incompatibilities section for details on Visual Basic .Net ActiveX object declarations.
4912085	AIX 5.2 /Sybase 12.5 Dispatch proxy in fnssl creates stacktraces

Table 6 Known Issues with Integration Server, EAI 3.1 *(Continued)*

Bug Number	Description
4912790	Fnscript commands documented as backbone are actually proxy level only. The FNscript commands `AddToJvmArgs`, `RemoveFromJvmArgs`, `AddToJmsClasspath`, and `RemoveFromJmsClasspath` are documented in the *Integration Server Backbone System Guide* as backbone level commands, when they only work at the proxy/component level. **Workaround:** Call the FNscript commands `AddToJvmArgs`, `RemoveFromJvmArgs`, `AddToJmsClasspath`, and `RemoveFromJmsClasspath` at the proxy/component level.
4913426	The documentation for SetProtocol FNscript command states the following for setting an ebXML proxy's protocol: `SetProtocol Name=ebXML ProcessName=`*procName* `| ProcessName=` The `ProcessName` argument is required. If the proxy is not intended to initiate a business process, the value for this argument should be left blank. `ProcessName=`*procName* specifies a process registered with the process engine that an Initiator proxy service will interact with. `ProcessName=` specifies that there is no process created by the service. Use this argument with Sender and Receiver services that interact with engine processes that are already started. The ProcessName= argument does not work as intended. **Workaround:** When specifying that there is no process created by the service, use `Process=` instead of `ProcessName=` to set the protocol.
4913444	`AddtoJvmArgs` and `RemoveFromJvmArgs` FNscript commands are not working correctly. **Workaround:** You can only add one JVM argument at a time using `AddToJvmArgs`. `RemoveFromJvmArgs` does not work. There is no workaround for this command.

How to Report Problems and Provide Feedback

If you have problems with Sun ONE Integration Server, EAI, contact Sun customer support using one of the following mechanisms:

- Sun Software Support services online at http://www.sun.com/service/sunone/software

This site has links to the Knowledge Base, Online Support Center, and ProductTracker, as well as to maintenance programs and support contact numbers.

- The telephone dispatch number associated with your maintenance contract

So that we can best assist you in resolving problems, please have the following information available when you contact support:

- Description of the problem, including the situation where the problem occurs and its impact on your operation
- Machine type, operating system version, and product version, including any patches and other software that might be affecting the problem
- Detailed steps on the methods you have used to reproduce the problem
- Any error logs or core dumps

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. Email your comments to Sun at this address:

docfeedback@sun.com

Please include the part number (817-4042-10) of the document in the subject line and the book title (*Integration Server, EAI 3.1 Release Notes*) in the body of your email.

Additional Sun Resources

Useful Sun ONE information can be found at the following Internet locations:

- Documentation for Integration Server, EAI.
  http://docs.sun.com/db/coll/S1_IntegrationServer_EAI_31
- Sun ONE Documentation
  http://docs.sun.com/prod/sunone
- Sun ONE Professional Services
  http://www.sun.com/service/sunps/sunone
- Sun ONE Software Products and Service
  http://www.sun.com/software

- Sun ONE Software Support Services
  http://www.sun.com/service/sunone/software
- Sun ONE Support and Knowledge Base
  http://www.sun.com/service/support/software
- Sun Support and Training Services
  http://www.sun.com/supportraining
- Sun ONE Consulting and Professional Services
  http://www.sun.com/service/sunps/sunone
- Sun ONE Developer Information
  http://sunonedev.sun.com
- Sun Developer Support Services
  http://www.sun.com/developers/support
- Sun ONE Software Training
  http://www.sun.com/software/training
- Sun Software Data Sheets
  http://wwws.sun.com/software

Copyright © 2003 Sun Microsystems, Inc. All rights reserved.

Sun, Sun Microsystems, the Sun logo, Solaris, Java and the Java Coffee Cup logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Use of Integration Server, EAI is subject to the terms described in the license agreement accompanying it.